Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работ №2 по курсу**

**«Операционные системы»**

Группа: М80 – 107Б-18

Студент: Лейченко Кира Андреевна

Преподаватель: Соколов

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2019.

**Содержание**

1. Постановка задачи
2. Общие сведения о программе
3. Общий метод и алгоритм решения
4. Основные файлы программы
5. Демонстрация работы программы
6. Вывод

**Постановка задачи**.

Программа позволяющая перенаправлять стандартный вывод. На вход программе подается имя выходного файла. Далее программа должна принимать различные команды интерпретатора команд и весь их вывод перенаправлять в файл.

**Общие сведения о программе**

Программа компилируется из одного файла lab\_oc1.c. В данном файле используются заголовочные файлы stdlib.h,unistd.h,fcntl.h,sys/wait.h В программе используются следующие системные вызовы:

1. **write** – для записи данных в файловый дескриптор
2. **open**-для открытия файла, определяемого по path-имени
3. **close** – для закрытия файла.
4. **fork** – для создания дочернего процесса.
5. **dup2**-для создания копии файлового дескриптора
6. **execl**-для замены текущего образа процесса новым образом процесса.
7. **wait**-приостанавливает выполнение текущего процесса до тех пор, пока дочерний процесс не завершится

**Общий метод и алгоритм решения**.

Для реализации поставленной задачи необходимо:

1. Используя системный вызов fork создается дочерний процесс, если пользователь ввел название файла, в противном случае программа завершает работу и выводит сообщение об ошибке.
2. Родительском процесс ожидает завершения работы дочернего процесса
3. Пока родительский процесс ожидает, дочерний процесс открывает файл, поданный в аргументе «argv[1]»,если такого файла нет, он создается. Затем с помощью утилиты dup2 перенаправляется стандартный поток вывода в файл и вызывается команда execl, которая заменяет текущий образ процесса на образ интерпретатора команд командной строки. Работа интерпретатора записывается в файл.

**Основные файлы программы**.

**Файл lab\_oc1.c**

#include<stdlib.h>

#include<unistd.h>

#include <fcntl.h>

#include <sys/wait.h>

int **main**(int argc, const char\*\*argv[])

{

    if(argv[1])

    {

    pid\_t parent = **fork**();

    if(parent < 0)

    {

**write**(STDIN\_FILENO,"Error, sorry,problem with fork\n",30);

        return -1;

    }

    if(parent==0)

    {

        int file = **open**(argv[1],O\_WRONLY|O\_TRUNC| O\_CREAT ,S\_IRWXU);

        if(file==-1)

        {

**write**(STDIN\_FILENO,"Error, not open file\n",30);

            return -1;

        }

        else

        {

**dup2**(file,1);

**execl**("/bin/sh","/bin/sh",NULL);

**exit**(1);

        }

    }

    if(parent>0)

    {

**wait**(NULL);

    }

    }

    else{

**write**(STDIN\_FILENO,"Error, sorry,no file name\n",30);

    }

    return 0;

}

**Демонстрация работы программы.**

kira@kira-LIFEBOOK-E458:~/Загрузки$ gcc lab\_oc1.c

kira@kira-LIFEBOOK-E458:~/Загрузки$ ./a.out 46.txt

date

ls

echo "hello world"

kira@kira-LIFEBOOK-E458:~/Загрузки$ cat 46.txt

Сб окт 19 19:27:32 MSK 2019

1.txt

2.txt

34.txt

3.txt

46.txt

a.out

Complex.cpp

Complex.h

file

file.txt

flash\_player\_npapi\_linux.x86\_64(1).tar.gz

flash\_player\_npapi\_linux.x86\_64(2).tar.gz

flash\_player\_npapi\_linux.x86\_64.tar.gz

JDT-2.1.rar

jdt5.jar

JDT5.ZIP

lab\_oc1.c

LGPL

libflashplayer.so

Main.cpp

my\_ans.txt

oc\_lab.c

PipesTest.cpp

Po\_fanu.jdtp

readme.txt

test(1).c

test.c

usr

VirtualTuringMachine

VirtualTuringMachine.zip

Zayavlenie\_na\_materialnuyu\_podderzhku\_ostronuzhdayuschemusya\_studentu.docx

Zayavlenie\_na\_perechislenie\_na\_kartu\_MIR.docx

Сложение десятичных чисел.jdtp

total 61092

hello world

**Вывод**

Данная лабораторная работа была нелегка для меня. Два дня мое жизни прошли в компании ноутбука, но все было не зря, прога работает. Гуглить на англиском оказалось результативнее, чем на русском, оказывается man удобнаяя штука. Мне кажется, системные вызовы импользуемые мной в данной лабораторной работе достаточно интересные и полезные в дальнейшей работе.